

## Entrega 1

1/

- **Nombre:** Arena azul
- **Logo:**



- **Layout del Resort.**



**a. Reporte de visión**

Nuestro proyecto Arena Azul surge con el fin de transformar la manera en que tradicionalmente se gestiona un resort, la intención es facilitar la experiencia de todo el equipo de trabajo y principalmente de los clientes, mediante un software el cual será una herramienta sencilla, practica y fácil de usar ya que se llevará a cabo el desarrollo de un programa de consola pensado para que todos los procesos desde la planeación de la demanda, el registro de huéspedes y sus mascotas, hasta las cuentas finales de costos, ventas y ganancias se haga de manera rápida y clara, sin enredos técnicos.

**Objetivo:** que el equipo de trabajo y gestión del resort tenga a la mano en tiempo real, la información que necesita para tomar decisiones proactivas: cuántas habitaciones quedan, cuántas personas han llegado, cuántos cupos hay en turismo y alimentación, y cómo van las finanzas del día y de esta manera optimizar al máximo los recursos aún disponibles y crear nuevas estrategias y ofertas para que el resort se mantenga lleno y los clientes satisfechos.

**Beneficios:**

- Agilidad: todo se registra en minutos y queda listo para consultar o exportar en un archivo CSV.
- Control total: se sabe de inmediato la disponibilidad de habitaciones y servicios, evitando sobrecupo o desperdicio.
- Visión de negocio: los reportes de ventas, costos y ganancias facilitan pensar en estrategias para crecer y mejorar la experiencia de los huéspedes.

**b. Especificación de requisitos**

Requisitos funcionales y no funcionales del software.

**Requisitos funcionales:** definen las acciones específicas, comportamientos, y operaciones que el software debe ejecutar para satisfacer las necesidades del usuario final, en el caso de Arena azul queremos un software que facilite la estancia de los clientes y el equipo de gestión del resort permitiendo que los huéspedes vivan una grata experiencia desde la llegada hasta su salida con unos procesos de registro amigables, fáciles y rápidos.

**¿Qué debe hacer el software?**

- Registrar huéspedes con sus datos personales (nombre, identificación, fechas de entrada y salida).
- Registrar mascotas asociadas a un huésped (nombre, tipo de mascota, observaciones).
- Permitir cancelar o modificar una reserva.
- Consultar en tiempo real la disponibilidad de habitaciones (las que están preparadas y disponibles y las que están disponibles, pero sin preparar)
- Mostrar en tiempo real el tipo de habitaciones que se tiene disponible (simples, dobles o familiares)
- Registrar reservas y confirmar el estado (reservada, ocupada, libre).

- Gestionar el inventario y capacidad de servicios adicionales (alimentación, turismo, actividades).
- Calcular automáticamente los costos de hospedaje, alimentación y servicios adicionales en total por día.
- Generar reportes diarios de ocupación, ventas y ganancias.
- Mostrar estadísticas en tiempo real (número de huéspedes, habitaciones ocupadas, ingresos acumulados).
- Guardar y recuperar datos para no perder información al cerrar el programa.

**Requisitos no funcionales:** especifican criterios que pueden usarse para juzgar la operación del sistema, más allá de los comportamientos específicos. Esto incluye aspectos como el rendimiento, seguridad, usabilidad, fiabilidad, y compatibilidad.

### ¿Cómo debe comportarse el software?

- El software debe ser amigable al usuario, el formato de consola debe ser claro, y fácil de usar para personas sin mucho conocimiento previo.
- El tiempo de respuesta de las consultas no debe ser muy extenso con el fin de que la gestión sea ágil.
- El sistema debe poder usarse en todo momento durante la operación del resort.
- El programa debe ejecutarse en distintos sistemas operativos (Windows, Linux, macOS).
- El código debe estar organizado para que se puedan agregar nuevas funciones sin rehacer todo el sistema.
- La información de los huéspedes debe estar protegida y no ser visible para usuarios no autorizados.
- El sistema debe evitar pérdidas de información en caso de cierre inesperado.
- Debe ser posible ampliar el sistema en el futuro (por ejemplo, integrar pagos en línea).

### c. Librerías

Para que Arena Azul funcione de manera práctica, sencilla y eficiente, vamos a apoyarnos en varias librerías de Python que nos facilitan desde el registro de huéspedes hasta el manejo de reportes.

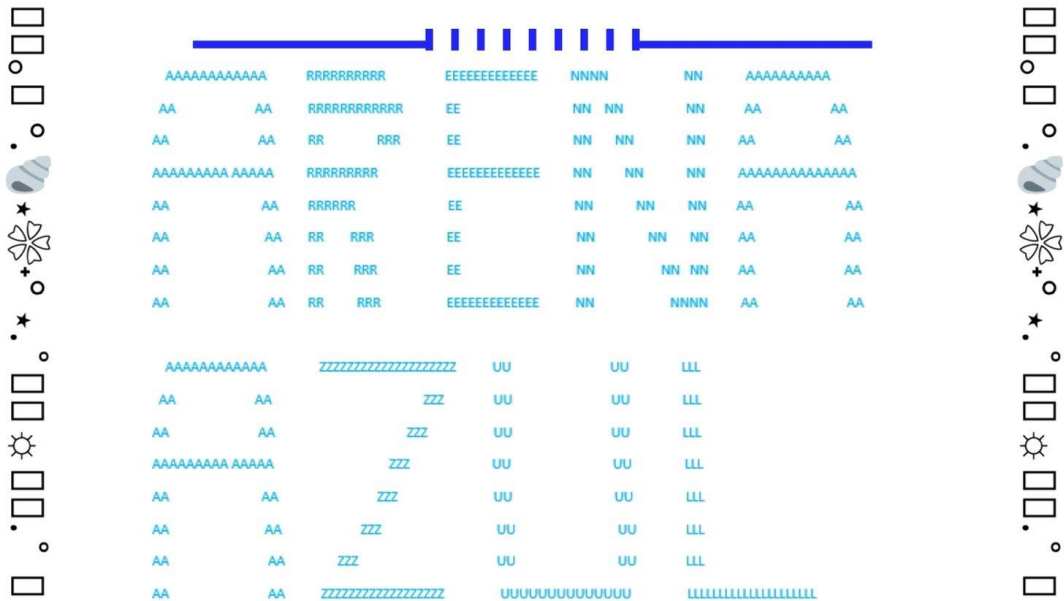
- **csv:** la usamos para guardar y leer los registros diarios en un formato que cualquiera del equipo de administración pueda abrir en Excel. Así, al final de cada día, el sistema guarda automáticamente quién llegó, qué habitación ocupó y el balance de ingresos.
- **datetime:** con esta mantenemos las fechas y horas bien organizadas. Cada vez que alguien hace check-in, el sistema marca la hora exacta y también nombra los archivos de reporte con la fecha, para que no se pierda nada en el historial.
- **getpass:** sirve para que el personal autorizado entre a la zona de administración sin que la contraseña quede visible. Si alguien necesita ver el reporte de ventas o la ocupación, solo el que tenga clave podrá acceder.
- **re:** nos ayuda a validar lo que escribe el usuario. Por ejemplo, cuando alguien registra un huésped, se asegura de que el documento de identidad tenga solo números y que el nombre no traiga caracteres raros que después den problemas.
-

- **pandas:** cuando queramos ver cómo se mueve la ocupación o comparar los ingresos de una semana frente a otra, esta librería nos permitirá hacer cálculos rápidos y tener los datos listos para analizarlos.
- 
- **matplotlib :** cuando llegué el momento de mostrar gráficas de temporadas altas o bajas, esta será la que ilustré los datos de forma clara y entendible para la gerencia.

Con estas herramientas, Arena Azul no solo registra y guarda la información diaria, también deja lista la base para ir creciendo; resolvemos el día a día y mañana podremos estudiar tendencias y tomar mejores decisiones para el resort.

#### **d. Visual**

## Bienvenidos



## Resort familiar

""Sistema de gestión""

-----Arena Azul-----

\*\*\*\*\*

*Modulo Administrador*

1. Ingrese usuario
2. Ingrese contraseña



""Clientes""

1. Registrar nuevo cliente
2. Consultar información de cliente
3. Actualizar datos de cliente
4. Eliminar cliente
5. Volver al menú principal

Espacios para toda la familia



Habitaciones confortables Senderos naturales Acceso a la playa

e. Algoritmos

<C:\Users\ISABELA\Documents\Pseudocodigo.txt>

d. Estructuras de datos

En el desarrollo de nuestro software para el resort Arena Azul, es viable utilizar diferentes estructuras de datos en Python que permiten organizar y gestionar la información de manera sencilla y eficiente.

**Listas:** Las listas son muy útiles porque permiten almacenar y modificar información de manera dinámica. Esto es clave en nuestro caso, ya que los huéspedes entran y salen constantemente y la información debe actualizarse. Por ejemplo, en una lista se puede guardar el nombre de los clientes, los datos generales de cada huésped o un listado de reservas activas.

**Tuplas:** Las tuplas, al igual que las listas, permiten guardar varios elementos en una sola variable, pero tienen la particularidad de que no se pueden modificar después de crearse. Esto resulta práctico para información que no debe cambiar, como las fechas de entrada y salida de una reserva o el número de habitación asignado.

**Diccionarios:** Los diccionarios son una manera de organizar la información por medio de pares de clave y valor. Esto significa que a cada dato importante se le puede asociar un identificador único. Son especialmente útiles en la administración del resort, ya que permiten almacenar datos de clientes, información de las habitaciones, gastos y precios de servicios como alimentación, habitaciones o tours. Incluso se pueden usar para generar contraseñas de acceso mediante la identificación de cada cliente.

De esta forma, listas, tuplas y diccionarios se convierten en herramientas esenciales para que el sistema de Arena Azul funcione de manera ordenada, clara y práctica, facilitando tanto la gestión diaria como el control de la información del resort.